International Journal of Swarm Intelligence Research
Volume 15 • Issue 1

A Novel Hybrid Binary Bat Algorithm for Global Optimization

Huijun Liang, Hubei Minzu University, China*
https://orcid.org/0000-0001-7911-6724

Aokang Pang, Hubei Minzu University, China
https://orcid.org/0000-0003-0902-8627

Chenhao Lin, Hubei Minzu University, China
https://orcid.org/0000-0002-9327-9260

Jianwei Zhong, Hubei Minzu University, China

ABSTRACT

In this article, a novel hybrid binary bat algorithm named HBBA is proposed for global optimization problems. First, to avoid simultaneous updating of bat velocity's dimensional components, i.e., elements of velocity vector, a random black hole model is modified to adapt to binary algorithm for updating in unknown spaces for each dimensional component individually. Through this way, the search ability of bats around the current group best is increased greatly. Second, a time-varying v-shaped transfer function, rather than a time-invariant one as in closely related works, is proposed to map velocity in continuous search space to a binary one. This accelerates the speed to switch individuals' positions, i.e., solutions in binary space. Third, a chaotic map is utilized to replace monotonous parameters in original binary bat algorithm, which is beneficial for avoiding premature convergence. Simulation results demonstrate the effectiveness of the proposed algorithm by three types of benchmark functions and unit commitment problem.

KEYWORDS

Binary Bat Algorithm, Random Black Hole Model, Transfer Function, Composition Benchmark Function, Unit Commitment

INTRODUCTION

Metaheuristic optimization algorithms are typically used to solve some complex optimization problems including nonconvex and nonlinear ones, which generally cannot be well solved by conventional mathematical methods. Although the solution generated by heuristic algorithms may not be equal to the exact optimal one, it is generally acceptable for real world engineering optimization problems.

DOI: 10.4018/IJSIR.342098
*Corresponding Author

Therefore, heuristic optimization algorithms have gained considerable interests during the past few decades (Abualigah et al., 2021; Hashim & Hussien, 2022). Several excellent representatives include genetic algorithm (GA) (Holland, 1992), particle swarm optimization (PSO) (Kennedy & Eberhart, 1995), differential evolution (DE) (G. Wang et al., 2022), grey wolf optimizer (GWO) (Nadimi-Shahraki et al., 2022), harmony search (HS) (Abarajithan & Vijayarani, 2022), ant colony optimization (ACO) (Dorigo & Gambardella, 1997), and bat algorithm (BA) (X. Yang, 2010; Akila & Christe, 2022).

Nearly all heuristic optimization algorithms proposed at the beginning are devoted to solving continuous variable optimization problems. However, many optimization problems in reality have discrete binary search space such as feature selection (El-Kenawy et al., 2022), 0–1 knapsack problem (Du et al., 2023), and unit commitment problem (Reddy et al., 2018). Therefore, some binary optimization algorithms are proposed according to their corresponding continuous versions to deal with binary optimization problems. For example, a sine cosine hybrid optimization algorithm with modified whale optimization algorithm (SCMWOA) was proposed by El-Kenawy et al. (2022). Its aim was to take advantage of WOA and SCA to solve problems with continuous and binary decision variables. An artificial algae algorithm's binary version (Turkoglu et al., 2022) was put forward to solve optimal attribute set for classification algorithms. A new binary multi-objective grey wolf optimizer was applied to dimensionality reduction problem in classification by Al-Tashi et al. (2020). By comparing and analyzing eight transfer functions including V-shaped and S-shaped, a binary quilibrium optimization algorithm was proposed by Abdel-Basset et al. (2021). A novel binary DE algorithm based on Taper-shaped transfer function (He et al., 2022) was proposed for solving knapsack problem and uncapacitated facility location problem. Besides, several other binary algorithms were proposed (Hichem et al., 2022; Pashaei & Pashaei, 2022; Usman et al., 2022) to solve feature selection problems. Although binary algorithms are proposed on the basis of continuous ones, there exist essential differences between them. Particularly, a transfer function is always required to map continuous space to a binary one in binary algorithm.

In the existing related works, the commonly used transfer functions are sigmoid function and its variants, both called S-shaped function (El-Kenawy et al., 2022; Pashaei & Pashaei, 2022). However, the application of S-shaped transfer function may slow down the convergence speed of algorithms because this type of function forces individuals to take values in 0 or 1. This means that the position will keep unchanged when speed increases. To overcome this drawback, V-shaped transfer function was proposed. The advantage of binary algorithms based on it is that they do not need to force individuals to take values in 0 or 1. More specifically, positions will keep unchanged if the corresponding velocity values are low, and will be replaced by their complements if the corresponding velocity values are high (Mirjalili & Lewis, 2013). This characteristic accelerates an individual's position change when search speed is changed (Usman et al., 2022; Mirjalili et al., 2014). Although algorithms with V-shaped transfer functions always show better performance than those with S-shaped transfer functions, the former are easy to suffer from local optima or premature convergence since the intrinsic drawbacks of original continuous algorithms are inherited by the binary ones. To further improve the diversity of populations, a time-varying mirrored S-shaped transfer function was proposed by Beheshti (2020) to help particles escape local optima. The performance of this method was demonstrated to be superior to S-shaped and V-shaped transfer functions based on methods by several benchmark functions.

Binary bat algorithm (BBA) was first proposed based on continuous BA (Mirjalili et al., 2014). There are lots of artificial bats in BA or BBA whose objective is to find an optimal solution for an optimization problem. Each artificial bat has some properties, such as velocity and position just as in PSO and GWO. Moreover, some unique properties, such as frequency, also exist in artificial bats. In BBA, some rules about velocity and position updating of artificial bats are carried out to adapt to binary problems. Typically, V-shaped transfer function is introduced to map continuous value space to a binary one, and the position (which is a binary value of artificial bat) is updated according to the corresponding velocity vector (which is a continuous value). The newly generated position is then regarded as the next base until the convergence rules are trigged. Based on BBA, several variants are developed in current related studies. For example, a binary cooperative BA (Zhang, 2014) was developed by integrating a weight parameter in velocity vector, and the transfer function was selected from four different V-shaped and S-shaped functions. An optimized binary BA (Gupta et al., 2019) was designed for feature selection. Remarkably, there are several drawbacks existing in original BBA and its variants:

(a) The update method of velocity vector weakens the algorithm's performance. In BBA, part of the dimensions in velocity vector are simply replaced by those randomly chosen from current group best. This means a further exploration or exploitation is lost in each iteration, especially for the search in unknown spaces.
(b) The switching speed of individuals' positions can be further increased. As stated previously, the time-varying S-shaped transfer function shows much better performance compared with S-shaped and V-shaped ones. This motivates us to further improve the performance of BBA.
(c) BBA is easy to trap into local optima and, thus, may suffer from premature convergence. There exist two monotonous parameters in BBA, i.e., pulse emission rate and loudness, which decrease the probabilities of generating new solution and accepting new solution, respectively.

In this paper, a new hybrid binary bat algorithm named HBBA is proposed, which fully considers the above disadvantages. Compared with BBA, the following improvements are achieved:

1. *The random black hole model* (J. Zhang et al., 2008) *is modified and integrated into HBBA*. Random black hole model is introduced and modified to realize the update in unknown spaces for each dimensional component of velocity vector separately, instead of directly substituting part of the dimensions by current group best as in BBA.
2. *A new time-varying V-shaped transfer function is proposed*. Compared with V-shaped transfer function, a time-varying one provides a faster switching speed for individuals' positions, i.e., values in binary space. This makes the proposed transfer function suitable for more complex composition functions (J. Liang et al., 2005).
3. *To avoid premature convergence, a chaotic map is introduced in HBBA*. Due to its nonrepetition characteristic, a chaotic map is introduced in HBBA to replace monotonous parameters to mitigate premature convergence problem.

The remainder of this paper is organized as follows. Section 2 describes original BA and BBA briefly. Section 3 explores the approaches to improve the performance of BBA for binary problems and proposes a new binary algorithm, HBBA. Section 4 gives the parameters sensitive analysis of the proposed algorithm and demonstrates the performance of HBBA by some experiments based on unimodal, multimodal, composite benchmark functions, and an engineering optimization problem. Concluding remarks and several future directions are given in Section 5.

ORIGINAL CONTINUOUS AND BINARY BAT ALGORITHM

This section briefly describes original principles of BA and BBA and gives the pseudo code of them.

Original Bat Algorithm

BA is proposed based on the hunting behavior of bats. Generally, nearly all bats use echolocation to detect obstacles or prey and distinguish directions. For example, when bats look for prey, they always fly randomly with a slow emission rate and a high loudness of acoustic pulses at first. Once a prey locates in some bat's hunting range, the emission rate will be increased while the loudness is decreased, and the pulse can keep going for a few milliseconds. This process is repeated until the prey is captured.

Inspired by the behavior of bats, the pulse frequency f, speed v and position x of bats at time t are mathematically described as follows:

$$f_i = f_{\min} + (f_{\max} - f_{\min})\kappa \tag{1}$$

$$v_i^{t+1} = v_i^t + (x_i^t - G_b^t) f_i \tag{2}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{3}$$

where $f_{\max}$ and $f_{\min}$ are the maximum and minimum frequency of pulse, respectively; κ is a random number in (0,1) drawn from a uniform distribution; G_b^t is the global best at time t.

In addition, a random walk is defined for local search to provide more diversity:

$$x_{i,new}^{t+1} = G_b^t + \sigma \overline{A} \tag{4}$$

where σ is a random number in (-1,1) drawn from a uniform distribution, and $\overline{A}$ is the average value of loudness for all bats.

To model the change of pulse emission rate r and loudness A, the following rules are given:

$$r_i^{t+1} = r_i^0 (1 - \exp(-\theta t)) \tag{5}$$

$$A_i^{t+1} = \alpha A_i^t \tag{6}$$

where r_i^0 is the initial pulse emission rate, θ and α are both constants in (0,1). The pseudo code of BA is listed as follows:

Algorithm 1: Bat algorithm

Input: Loudness A_i, frequency f_i, pulse rates r_i, population size n and max iterations N_{gen}

Output: The optimal solution G_b^t and best fitness f_b

1: Get fitness values of all individuals and calculate current group best

2: **while** $t < N_{\text{gen}}$ **do**

3: **while** $i \le n$ **do**

```
            Generate new solution by adjusting frequency and velocity according to Eq. (1)-(3)
            if rand > r_i then
                Select a solution among the best solutions
                Generate a new solution using Eq. (4)
            end if
            Generate new fitness value f_new
            if rand < A(i) and f_new < f(G_b^t) then
                Accept the new solution
                Change r and A according to Eq. (5) and Eq. (6)
            end if
            Rank solutions and find the optimal solution G_b^t
        end while
    end while
    Present the final solution
```

Original Binary Bat Algorithm

BA is only suitable to continuous problems. However, in binary space, algorithms must be faced with the condition that there are only two values, i.e., 0 and 1 in binary domain. Therefore, the position updating rule Eq. (3) cannot work anymore in binary space, and the relation must be found to connect individuals' velocities and positions.

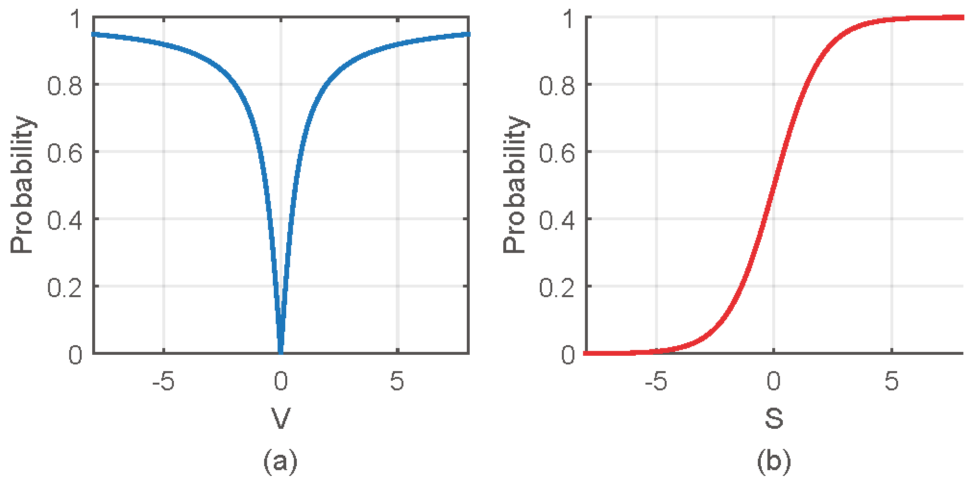
Position updating in binary space can only change in 0 or 1. In binary algorithms, this change should be accomplished by individuals' velocities. The key lies in how a continuous value (i.e., velocity of individual) can be used to update the position in binary space. Transfer function is a commonly used way to deal with this issue (Aslan et al., 2019). In BBA, a V-shaped transfer function is proposed as follows:

$$V(v_i^k(t)) = \left|\frac{2}{\pi} * \arctan(\frac{\pi}{2} * v_i^k(t))\right| \tag{7}$$

$$x_i^k(t+1) = \begin{cases} x_i^k(t)^{-1}, if \text{ rand} < V(v_i^k(t+1)), \\ x_i^k(t), \quad if \text{ rand} \geq V(v_i^k(t+1)), \end{cases} \tag{8}$$

where $x_i^k(t)$ denotes the i-th artificial bat at iteration t in the k-th dimension; $x_i^k(t)^{-1}$ means the complement of $x_i^k(t)$, and rand is a random number in (0,1). The figure of V-shaped transfer function is shown in Figure 1(a). Compared with sigmoid transfer function (Figure 1[b]), which is used in other algorithms, V-shaped transfer function provides individuals higher velocity to change their positions in binary space. Through this method, the search process of continuous search space can be mapped to a binary one. This is also the main difference between BA and BBA. The pseudo code of BBA is shown in Algorithm 2.

Figure 1. (a) V-Shaped transfer function in BBA, (b) Sigmoid transfer function



Algorithm 2: Binary bat algorithm

Input: Loudness A_i, frequency f_i, pulse rates r_i, population size n and max iterations N_{gen}

Output: The optimal solution G_b^t and best fitness f_b

Get fitness values of all individuals and calculate current group best

while $t < N_{\text{gen}}$ **do**

  while $i \le n$ **do**

    Generate new solution by adjusting frequency and velocity according to Eq. (1)-(2)

    Calculate V-shaped transfer function value by Eq. (7)

    Update solutions by Eq. (8)

    if $rand > r_i$ **then**

      Select G_b^t among the current best solutions

      Change some dimensions in v based on G_b^t

    end if

    if $rand < A(i)$ and $f_{\text{new}} < f(G_b^t)$ **then**

      Accept the new solution

      Change r and A according to Eq. (5) and Eq. (6)

    end if

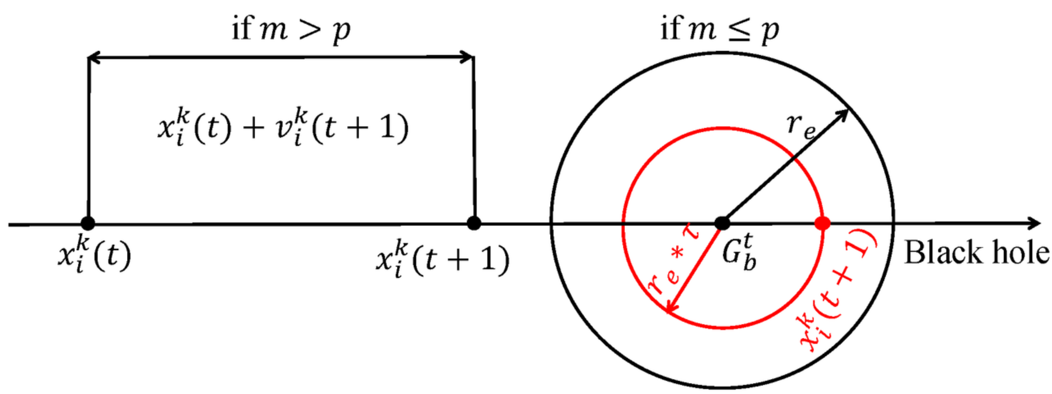
    Rank solutions and find the optimal solution G_b^t

  end while

end while

Present the final solution

Figure 2. Schematic of random black hole model



THE PROPOSED BINARY ALGORITHM

A new hybrid binary bat algorithm named HBBA is proposed in this section. Specifically, the following steps are improved compared with original BBA: First, individuals' exploration and exploitation are enhanced by integrating random black hole model (H. Zhang et al., 2020). Second, to cope with the challenge of composition test functions (Gupta et al., 2019), a new time-varying V-shaped transfer function is proposed. Third, a chaotic map is introduced to improve the diversity of HBBA.

Random Black Hole Model

A black hole in a galaxy can come from any mass that collapses down to the Schwarzschild radius (Bambi et al., 2019). Theoretically, the escape speed in a black hole is equal to the speed of light. However, since no object can go faster than light, any object located in the effective range of a black hole will be absorbed by it.

According to a real black hole, a random black model is proposed by J. Zhang et al. (2008), where each particle in PSO is treated as a star, and the corresponding fitness value is gravity. Thus, the position of a particle is influenced by the gravity of group best and local optimum at every iteration. Generally, the real global optimum is unknown. Therefore, the current group best is treated as the base point for generating a black hole in each iteration.

This idea is adopted in RCBA (H. Liang et al., 2018), and the schematic is shown in Figure 2. As seen in the figure, r_e represents the effective radius of the base point (it means that a new black hole will be generated in this range); G_b^t is the base point, i.e., the current group best; p is a constant threshold value located in (0,1), and m is a random number between 0 and 1. If the randomly generated number m is not greater than p, a new random black hole is generated around G_b^t in $(-r_e, +r_e)$. Then, the new solution is updated as follows:

$$x_i^k(t+1) = G_b^t + r_e * \tau \tag{9}$$

where $\tau \in [-1, +1]$ obeys uniform distribution, and $x_i^k(t+1)$ is the k-th dimension in the i-th individual at iteration $t+1$. If $m > p$, the position is updated according to Eq. (3).

However, the above process is only suitable for continuous problems because the position $x_i^k(t+1)$ is changed in continuous domain. For discrete binary problems, the original random black hole model should be modified. As analyzed in Section 2, the rule for updating position, i.e., Eq. (3), is not suitable in binary space anymore, and the position updating must depend on the velocity's change of artificial bat.

Based on the above analysis, the following rule is performed immediately once velocity update rule Eq. (2) is executed:

$$v_i^k(t+1) = G_b^k(t) + r_e * \tau \tag{10}$$

The update process for a modified random black hole model in this study is described in Algorithm 3.

Algorithm 3: Update process for a modified random black hole model

Input: G_b^t, p and r_e
Output: v_i^k
1: **for** each dimension in v_i^k **do**
2: Generate a random value for u
3: **if** $u < p$ **then**
4: Update a dimension for $v_i^k(t+1)$ by Eq. (10)
5: **end if**
6: **end for**

The new update rule for velocity, i.e., Eq. (10), takes several superiorities:

(a) Compared with Eq. (2), Eq. (10) can realize each dimension's updating in velocity vector v separately. Furthermore, as seen in line 3 of Algorithm 3, Eq. (10) is only executed when $u < p$. It means that only part dimensions are updated in each iteration step. This characteristic is beneficial for improving global search ability of artificial bats and providing more diversity of solutions.
(b) The ability of exploration and exploitation for artificial bats is enhanced. At the earlier stage of algorithm's iteration, r_e is assigned with a relatively big value. This brings a wider search space in preliminary stage of iterations and, hence, the ability of exploration is enhanced. As iteration goes on, a relatively good solution is obtained, and a small search space should be provided to get more precious solution around current group best. One way is to decrease the effective radius r_e and, thus, the ability of exploitation is also enhanced.

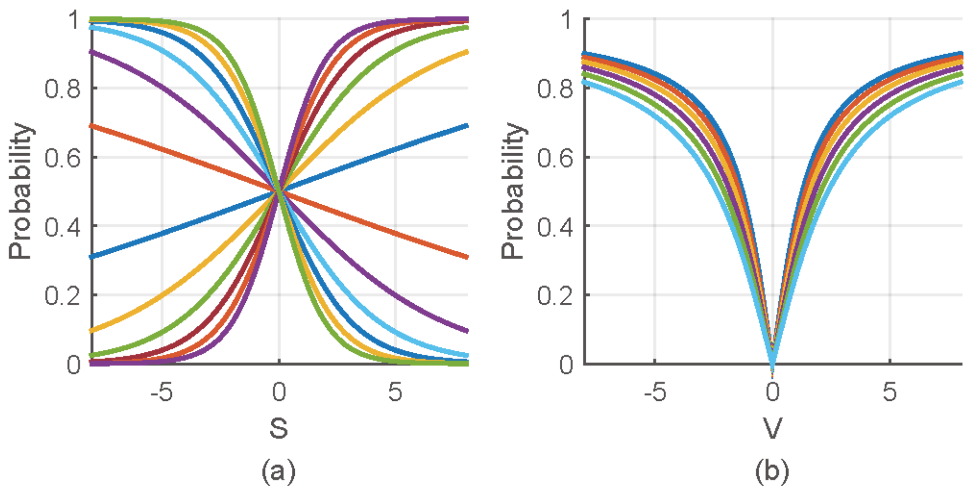
Time-Varying V-Shaped Transfer Function

The velocity v generated by Eq. (10) is a continuous value and must be mapped into a binary space. The commonly used method is to introduce transfer function into algorithms, such as S-shaped, V-shaped, and linear normalized transfer function. However, algorithms with the three types of transfer functions show slow convergence speed and sometimes may trap into local optima (Beheshti, 2020).

To overcome the above shortcoming, a time-varying mirrored sigmoid transfer function (Beheshti, 2020) is introduced, as shown in Figure 3(a). As seen in this figure, two sigmoid functions are introduced as follows:

$$S(v_i^k(t+1), \eta_t) = 1 / (1 + e^{\eta_t(-v_i^k(t+1))}) \tag{11}$$

Figure 3. (a) Time-Varying mirrored sigmoid transfer function, (b) Proposed time-varying v-shaped transfer function



$$S'(v_i^k(t+1),\eta_t) = 1 / (1 + e^{\eta_t(v_i^k(t+1))}) \tag{12}$$

where η_t is a time varying parameter, and $k = 1 \cdots d$ in binary space (d is the dimensions of a solution in binary space). η_t is assigned a relatively big value at first, and gradually decreased. By this means, a strong exploration is achieved at the beginning of iteration, and a well exploitation is obtained as strong exploration is achieved at the beginning of iteration, and a well exploitation is obtained as iteration goes on.

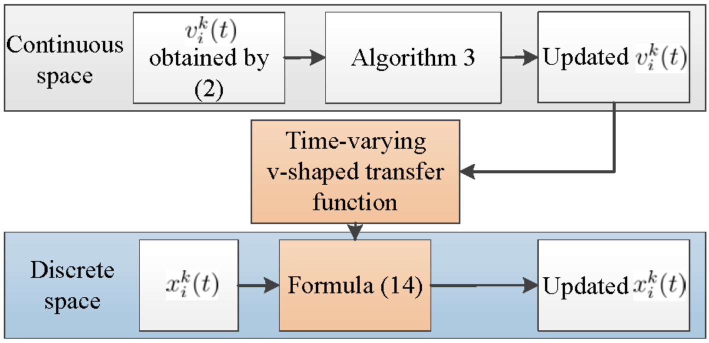
Inspired by the time-varying mirrored S-shaped transfer function, a time-varying V-shaped transfer function is proposed in this paper. This idea comes from the fact that a V-shaped transfer function provides a higher speed to switch individuals' positions compared with an S-shaped transfer function (Mirjalili et al., 2014). Therefore, a time-varying V-shaped transfer function is generated as below:

$$V'(v_i^k(t+1)) = \left| \frac{\pi}{2} \arctan\left(\frac{2}{\pi} \frac{v_i^k(t)}{1 + e^{\theta_t}}\right) \right| \tag{13}$$

$$x_i^k(t+1) = \begin{cases} x_i^k(t)^{-1}, \mathit{if} \ \text{rand} < V'(v_i^k(t+1)), \\ x_i^k(t), \quad \mathit{if} \ \text{rand} \geq V'(v_i^k(t+1)), \end{cases} \tag{14}$$

where θ_t is a time varying parameter as iteration goes on. This way, artificial bats are forced to search in binary space. Figure 3(b) shows the curves of a proposed time-varying V-shaped transfer function. Compared with a time-varying mirrored S-shaped transfer function, the proposed time-varying V-shaped transfer function provides not only a faster switching speed but also a strong exploration and exploitation for individuals, and it is especially suitable for complex optimization problems (e.g., composite benchmark functions [J. Liang et al., 2005]).

Figure 4. The schematic of mapping a continuous space to a binary one



Chaotic Map

In mathematics, a chaotic map exhibits some sort of chaotic behavior. Chaotic systems are generally characterized by their intrinsic stochastic properties, which have nonrepetition characteristics and an ergodic nature (H. Liang et al., 2018; Suresh & Lal, 2017). These features determine that chaotic maps are especially suitable to replace some parameters in metaheuristic algorithms. This way, the diversity of algorithms is increased and, hence, the premature convergence is mitigated or even avoided. A chaotic map has been successfully applied to many heuristic algorithms (Hematpour & Ahadpour, 2021; Qin et al., 2017).

There are two monotonous parameters, i.e., pulse emission rate r and loudness A in BBA or BA. As seen in Eq. (5) and Eq. (6), r is a monotonically increasing sequence and A is a monotonically decreasing one. Indeed, the above features limit the performance of BA (see lines 5 and 10 in Algorithm 1) and BBA (see lines 7 and 11 in Algorithm 2). For example, in BBA, a new solution can be accepted when one condition $rand > r_i$ is satisfied (see line 11 in Algorithm 2). However, since A is a monotonically decreasing parameter, the chance to accept new solution is decreased as iteration goes on. Therefore, some relatively acceptable solutions may be missed. To overcome this shortcoming, a chaotic map is introduced to replace r and A to increase the diversity of the algorithm.

The Proposed Binary Algorithm

By integrating the modified random black hole model and the proposed time-varying V-shaped transfer function, artificial bats have the ability to find more acceptable solutions in binary search space. The schematic of mapping continuous variable v to a discrete value x in binary space is shown in Figure 4.

The pseudo of proposed binary algorithm HBBA is shown in Algorithm 4, and the underlined parts are the improvements, which are different from BBA.

Algorithm 4: The proposed binary bat algorithm (HBBA)

Input: Loudness A_i, frequency f_i, pulse rates r_i, population size n and max iterations N_{gen}

Output: The optimal solution G_b^t and best fitness f_b

1: Get fitness values of all individuals and calculate current group best

2: **while** $t < N_{\text{gen}}$ **do**

3: **while** $i \le n$ **do**
4: Generate new solution by adjusting frequency and velocity according to Eq. (1)-(2)
5: **if** $rand > r_i$ **then**
6: Update velocity v by the modified random black hole model (i.e., Algorithm 3)
7: **end if**
8: Calculate time-varying V-shaped transfer function by Eq. (13)
9: Update solutions by Eq. (14)
10: **if** $rand < A(i)$ and $f_{\text{new}} < f(G_b^t)$ **then**
11: Accept the new solution
12: Change the values of r and A by chaotic map
13: **end if**
14: Rank solutions and find the optimal solution G_b^t
15: **end while**
16: **end while**
17: Present the final solution

The main process is addressed as follows:

(a) Initialize parameters of HBBA including loudness A_i, pulse emission rate r_i, pulse frequency $f_{\min}$ and $f_{\max}$. The initialized velocities of populations are all set to 0. The positions of populations are random initialized in {0, 1} which are shown as below:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} x_1^1 & x_1^2 & \cdots & x_1^k \\ x_2^1 & x_2^2 & \cdots & x_2^k \\ \vdots & \vdots & \vdots & \vdots \\ x_n^1 & x_n^2 & \cdots & x_n^k \end{bmatrix} \tag{15}$$

where the length of each dimension x_i^k should be defined according to different problems.

(b) Get fitness values according to the above initialized positions and find the current group best G_{b}^t.
(c) At the beginning of iterations, velocity is first updated by Eq. (2), and then is further improved by the modified random black hole model (see lines 6-9). This way, each dimension in v_i^k is updated separately.
(d) Update positions by the proposed time-varying V-shaped transfer function, i.e., Eq. (13) and (14). This step maps the continuous value v_i^k to the binary solution x_i^k.
(e) If the condition in line 12 is satisfied, the new generated solution is accepted. Pulse emission rate r and loudness A are respectively updated by chaotic map instead of by Eq. (2) and (3). This is helpful for improving the diversity of solutions, and thus mitigating premature convergence.

Table 1. Unimodal benchmark functions

Function	Range	$f_{\min}$
$f_1(\mathbf{x}) = \sum_{i=1}^{k} x_i^2$	[-100,100]	0
$f_2(\mathbf{x}) = \sum_{i=1}^{k} \|x_i\| + \prod_{i=1}^{k} \|x_i\|$	[-10,10]	0
$f_3(\mathbf{x}) = \sum_{i=1}^{k} (\sum_{j=1}^{i} x_j)^2$	[-100,100]	0
$f_4(\mathbf{x}) = \max_{i=1,\ldots,k} \|x_i\|$	[-100,100]	0
$f_5(\mathbf{x}) = \sum_{i=1}^{k-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	[-30,30]	0
$f_6(\mathbf{x}) = \sum_{i=1}^{k} (x_i + 0.5)^2$	[-30,30]	0
$f_7(\mathbf{x}) = \sum_{i=1}^{k} i x_i^4 + \text{random}[0,1)$	[-1.28,1.28]	0

EXPERIMENTAL RESULTS

Three types of benchmark functions are employed to evaluate the superiority of the proposed binary algorithm HBBA in this section, including unimodal, multimodal, and composite benchmark functions (J. Liang et al., 2005; X. Yang, 2010), which are given in Tables 1 to 3, respectively. The description of composition functions is given in J. Liang et al. (2005), which interested readers can refer to for details. In addition, a unit commitment problem further validates the effectiveness of the proposed HHBA in an engineering optimization problem. The positions, i.e., solutions, are expressed by binary numbers with 16 bits in each dimension, and the first bit represents the symbol ('0' denotes a positive number and '1' denotes a negative number). The following gives the simulation results for the above three types of benchmark functions. Because BBA achieves better solutions compared to BPSO and GA (Mirjalili et al., 2014), this paper puts the emphasis on the comparison between HBBA and BBA. This simulation is executed in MATLAB on INTEL i5-11300H, 3.10GHz, and 16 GB RAM.

Parameter Sensitive Analysis

In order to improve the reliability and generality of the proposed algorithm, this subsection focuses on the sensitivity analysis of the algorithm parameters that are required to be analyzed, which are the population size n, max iterations N_{gen} and the effective radius of random black role model r_e for the proposed algorithm from Algorithm 4. The values of $f_{\min}$ and $f_{\max}$ are set to 0 and 2 from the BBA (Mirjalili et al., 2014). As for the modified random black hole model, threshold p is assigned as 0.5. Besides, the unimodal function f_5, f_6, multimodal function f_8, f_{10}, and composition function

Table 2. Multimodal benchmark functions

Function	Range	$f_{\min}$
$f_8(\mathbf{x}) = \sum_{i=1}^{k} -x_i \sin(\sqrt{\mid x_i \mid})$	[-500,500]	-418.9829*5
$f_9(\mathbf{x}) = \sum_{i=1}^{k} [x_i^2 - 10\cos(2\pi x_i) + 10]$	[-5.12,5.12]	0
$f_{10}(\mathbf{x}) = -20e^{(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{k} x_i^2})} - e^{(\frac{1}{k}\sum_{i=1}^{k}\cos(2\pi x_i))} + 20 + e$	[-32,32]	0
$f_{11}(\mathbf{x}) = \frac{1}{4000}\sum_{i=1}^{k} x_i^2 - \prod_{i=1}^{k}\cos(\frac{x_i}{\sqrt{i}}) + 1$	[-600,600]	0
$f_{12}(\mathbf{x}) = -\sum_{i=1}^{k} \sin(x_i)\cdot(\sin(\frac{ix_i^2}{\pi}))^{2m}, m = 10$	[0, π]	-4.687
$f_{13}(\mathbf{x}) = [e^{-\sum_{i=1}^{k}(x_i/\beta)^{2m}} - 2e^{-\sum_{i=1}^{k} x_i^2}]\prod_{i=1}^{k}\cos^2(x_i), m = 5$	[-20,20]	-1
$f_{14}(\mathbf{x}) = \left([\sum_{i=1}^{k}\sin^2(x_i)] - e^{-\sum_{i=1}^{k} x_i^2}\right) e^{-\sum_{i=1}^{k}\sin^2\sqrt{\mid x_i \mid}}$	[-10,10]	-1

f_{17} , f_{18} are decided to be representative functions for parameter sensitive analysis. The following 'Ave', 'SD', 'Med' and 'Time' denote the average value, standard deviation, median, and average execution time, respectively. The specific analyses of the above parameters are as follows.

The population, as one of the most important parameters in the algorithm, has a significant impact on the convergence and robustness of the algorithm. Young's research suggests that a population size between 15 and 50 would be appropriate for most issues (Mirjalili et al., 2014). In order to analyze the effect of population size on the convergence and robustness of the proposed algorithm, we set the population size to 10, 20, 30, 40, and 50 to test the representative function for 30 independent runs, as shown in Table 4. It can be seen from Table 1 that, as the population size grows, both the convergence and robustness of the algorithm perform better, except for f_{10} . Therefore, the best population size for the proposed algorithm is 50.

The number of iterations of the algorithm is also vital to the convergence results. Therefore, the iteration number of 100, 200, 300, 400, and 500 are employed to test the representative functions to determine which one is better for the proposed algorithm for 30 independent runs. It can be seen from Table 5 that the average convergence result and standard deviation of proposed algorithm for 500 iterations are the best among all the results of the comparison iteration experiments. Although the standard deviations of f_8 and f_{18} at 500 iterations are not best values for the proposed algorithm in all iterations, they also perform well. In terms of the results, the maximum iteration of 500 is suited for the proposed algorithm.

Table 3. Composite benchmark functions

Function	Definition	Coefficients	$f_{\min}$
f_{15} (g)	$g_1, g_2 \dots, g_{10} = \mathrm{SF}$	$[\sigma_1, \sigma_2, \dots, \sigma_{10}] = [1, 1, \dots, 1]$ $[\lambda_1, \lambda_2, \dots, \lambda_{10}] = [0.05, \dots, 0.05]$	0
f_{16} (g)	$g_1, g_2 \dots, g_{10} = \mathrm{GF}$	$[\sigma_1, \sigma_2, \dots, \sigma_{10}] = [1, 1, \dots, 1]$ $[\lambda_1, \lambda_2, \dots, \lambda_{10}] = [0.05, \dots, 0.05]$	0
f_{17} (g)	$g_1, g_2 \dots, g_{10} = \mathrm{RF}$	$[\sigma_1, \sigma_2, \dots, \sigma_{10}] = [1, 1, \dots, 1]$ $[\lambda_1, \lambda_2, \dots, \lambda_{10}] = [1, 1, \dots, 1]$	0
f_{18} (g)	$g_1, g_2 = \mathrm{AF}, g_3, g_4 = \mathrm{RF},$ $g_5, g_6 = \mathrm{WF}, g_7, g_8 = \mathrm{GF},$ $g_9, g_{10} = \mathrm{SF}$	$[\sigma_1, \sigma_2, \dots, \sigma_{10}] = [1, 1, \dots, 1]$ $[\lambda_1, \lambda_2, \dots, \lambda_{10}] = [5 / 32, 5 / 32, 1, 1, 5/0.5,$ $5/0.5, 5/100, 5/100, 5/100, 5/100]$	0
f_{19} (g)	$g_1, g_2 = \mathrm{RF}, g_3, g_4 = \mathrm{WF},$ $g_5, g_6 = \mathrm{GF}, g_7, g_8 = \mathrm{AF},$ $g_9, g_{10} = \mathrm{SF}$	$[\sigma_1, \sigma_2, \dots, \sigma_{10}] = [1, 1, \dots, 1]$ $[\lambda_1, \lambda_2, \dots, \lambda_{10}] = [1 / 5, 1 / 5, 5 / 0.5, 5 / 0.5, 5/100,$ $5/100, 5/32.5, 5/32.5, 5/100, 5/100]$	0
f_{20} (g)	$g_1, g_2 = \mathrm{RF}, g_3, g_4 = \mathrm{WF},$ $g_5, g_6 = \mathrm{GF}, g_7, g_8 = \mathrm{AF},$ $g_9, g_{10} = \mathrm{SF}$	$[\sigma_1, \sigma_2, \dots, \sigma_{10}] = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]$ $[\lambda_1, \lambda_2, \dots, \lambda_{10}] = [0.1 * 1 / 5, 0.2 * 1 / 5, 0.3 * 5 / 0.5, 0.4 * 5 / 0.5,$ $0.5 * 5 / 100, 0.6 * 5 / 100, 0.7 * 5 / 32, 0.8 * 5 / 32, 0.9 * 5 / 100, 5 / 100]$	0

From the description of the random black hole model in Section 3, it can be seen that r_e has a significant effect on the ability of exploration and exploitation for artificial bat. Thus, a suitable r_e is crucial for the performance of the proposed algorithm. According to the analysis of H. Liang (2018), we set r_e to 3 group of piecewise parameters, such as r_{e1}, r_{e2}, and r_{e3}, respectively. Besides, to further analyze the effect of r_e size on the convergence and robustness of the proposed algorithm, we set r_{e4}, r_{e5} as 0.1, and 0.01, respectively. The five group of r_e are applied to test on representative function for 30 independent runs, which are shown in Table 6. It can be clearly seen that the proposed algorithm performs best in unimodal and multimodal benchmark function f_5, f_6, f_8, and f_{10} when using r_{e4}. However, the proposed algorithm obtains the smallest average results and standard deviation using r_{e2} in the representative composite benchmark functions. Therefore, r_e in this paper is set to r_{e4} for unimodal, multimodal benchmark functions and r_{e2} for composite benchmark functions.

Simulations for Unimodal Benchmark Functions

This subsection gives the simulation results for unimodal benchmark functions, which are listed in Table 4. The dimension for each function is set to 5, and there are a total of 30 independent runs for each function. The statistical results are shown in Table 7, and the convergence processes of best solutions for functions f_1 to f_7 are given in Figures 5 and 6.

Table 4. Population size analysis of representative functions for the proposed algorithm

Function	10	20	30	40	50
f_5 Ave	3.15E+00	3.03E+00	2.78E+00	2.75E+00	**2.68E+00**
SD	2.49E-01	2.41E-01	3.36E-01	2.11E-01	**2.00E-01**
f_6 Ave	5.96E-01	4.44E-01	2.94E-01	2.94E-01	**2.27E-01**
SD	1.50E-01	1.32E-01	1.16E-01	1.27E-01	**9.69E-02**
f_8 Ave	-1.30E+03	-1.42E+03	-1.55E+03	**-1.58E+03**	**-1.58E+03**
SD	1.44E+02	1.37E+02	1.31E+02	1.36E+02	**1.30E+02**
f_{10} Ave	**8.88E-16**	**8.88E-16**	**8.88E-16**	**8.88E-16**	**8.88E-16**
SD	**0**	**0**	**0**	**0**	**0**
f_{17} Ave	1.92E+02	1.77E+02	1.75E+02	1.66E+02	**1.55E+02**
SD	3.60E+01	3.82E+01	3.02E+01	3.43E+01	**2.99E+01**
f_{18} Ave	3.82E+02	3.62E+02	3.50E+02	3.53E+02	**3.38E+02**
SD	2.43E+01	1.83E+01	2.07E+01	1.51E+01	**1.50E+01**

Table 7 gives the comparison results among the proposed binary algorithm HBBA_1(only modified random black hole model), HBBA_2(modified random black hole model and time-varying V-shaped transfer function), HBBA(modified random black hole model, time-varying V-shaped transfer function and chaotic map), BBA, BPSO, GA and binABC (Kiran, 2015) for unimodal benchmark functions. Each simulation is executed in 30 independent runs. The data

Table 5. Max iteration analysis of representative functions for the proposed algorithm

Function	100	200	300	400	500
f_5 Ave	2.96E+00	2.95E+00	2.78E+00	2.75E+00	**2.68E+00**
SD	4.03E-01	3.11E-01	3.63E-01	**3.36E-01**	**3.36E-01**
f_6 Ave	4.70E-01	4.36E-01	3.44E-01	3.84E-01	**2.27E-01**
SD	1.21E-01	1.26E-01	1.12E-01	1.08E-01	**9.69E-02**
f_8 Ave	-1.39E+03	-1.43E+03	-1.45E+03	-1.48E+03	**-1.58E+03**
SD	**1.19E+02**	2.07E+02	1.40E+02	1.66E+02	1.30E+02
f_{10} Ave	**8.88E-16**	**8.88E-16**	**8.88E-16**	**8.88E-16**	**8.88E-16**
SD	**0**	**0**	**0**	**0**	**0**
f_{17} Ave	2.84E+02	2.19E+02	1.89E+02	1.77E+02	**1.55E+02**
SD	4.96E+01	3.40E+01	3.44E+01	3.72E+01	**2.99E+01**
f_{18} Ave	4.58E+02	3.87E+02	3.65E+02	3.56E+02	**3.38E+02**
SD	3.28E+01	1.79E+01	2.02E+01	**1.25E+01**	1.50E+01

Table 6. The effective radius of random black role model analysis of representative functions for the proposed algorithm

Function	r_{e1}	r_{e2}	r_{e3}	r_{e4}	r_{e5}
f_5 Ave	2.68E+00	3.49E+00	2.69E+00	**2.57E+00**	3.01E+00
SD	3.48E-01	3.33E-01	4.27E-01	**3.36E-01**	3.42E-01
f_6 Ave	2.73E-01	6.35E-01	2.69E-01	**2.27E-01**	4.36E-01
SD	1.12E-01	1.72E-01	1.13E-01	**9.69E-02**	1.63E-01
f_8 Ave	**-1.63E+03**	-1.62E+03	-1.58E+03	**-1.63E+03**	-1.46E+03
SD	1.66E+02	2.07E+02	1.40E+02	**1.19E+02**	1.30E+02
f_{10} Ave	**8.88E-16**	1.65E-02	**8.88E-16**	**8.88E-16**	**8.88E-16**
SD	**0**	3.49E-02	**0**	**0**	**0**
f_{17} Ave	1.88E+02	**1.64E+02**	1.81E+02	1.89E+02	2.02E+02
SD	2.73E+01	**2.40E+01**	2.81E+01	3.98E+01	3.52E+02
f_{18} Ave	3.39E+02	**3.28E+02**	3.49E+02	3.51E+02	3.38E+02
SD	4.62E+01	**2.59E+01**	2.98E+01	4.12E+01	2.96E+01

about BBA, BPSO and GA come from Mirjalili et al. (2014). As seen in Table 5, for unimodal functions f_1 to f_4, HBBA obtains the real global optimal solutions, i.e., 0, in all 30 independent runs. For BBA, BPSO, GA and bin ABC, the medians in 30 independent runs for functions f_1 to f_4 are 1.2037, 4.6684, 2.6534, 405.42, respectively. It means that the four algorithms cannot get the real global optimal solutions. As for f_5 to f_7, HBBA gets the smallest values in all the four items, i.e., Ave, SD, and Med compared with values obtained by BBA, BPSO, GA and binABC. For example, the medians obtained by HBBA are 2.6767, 0.2241, and 9.22e-5 for f_5 to f_7, respectively. These values are about 19.54%, 21.94%, and 0.74% compared with the corresponding values generated by BBA, respectively. For HBBA_1, HBBA_2 and HBBA, the average value and medians obtained by HBBA are best, and the average value and medians obtained by HBBA_2 are better than that of HBBA_1. It can be clearly demonstrated that every improvement to the BBA in this paper is effective.

The convergence curves of the best solutions for f_1 to f_7 are given in Figures 5 and 6. As seen in the figures, HBBA achieves excellent convergence characteristics. For example, the group bests at the 100-th iteration obtained by HBBA are about 0, 0, 0, 0, 2.338, 0.03573, and 0.0006704 for f_1 to f_7, respectively, while for BBA (the performance of BBA is superior to BPSO and GA according to [Mirjalili et al., 2014], so HBBA is only compared with BBA in this paper), the corresponding values are about 8.5, 0.47, 64.3, 5.9, 100.1, 9.9, and 0.012, respectively (Mirjalili et al., 2014). Meanwhile, the optimal solutions obtained by HBBA are also smaller than the corresponding values obtained by BBA. Therefore, the convergence performance of HBBA is superior to BBA totally.

Based on the above analysis, it can be concluded that HBBA not only achieves the best solutions but also obtains the best convergence performance compared to BBA, PSO, and GA for unimodal benchmark functions f_1 to f_7.

Table 7. Simulation results for unimodal benchmark functions in 30 independent runs

Function	HBBA_1	HBBA_2	HBBA	BBA	BPSO	GA	binABC
f_1 Ave	0.0020	0	**0**	1.8518	5.2965	10.0705	453.45
SD	0.0110	0	**0**	2.4981	2.7657	24.9445	234.14
Med	0.0052	0	**0**	1.2037	4.6684	2.6534	405.42
Time	3.1982	3.2581	2.8370	0.3838	0.5844	1.5705	**0.2811**
f_2 Ave	3.15E-04	0	**0**	0.0965	0.2292	0.2695	3.4323
SD	1.75E-03	0	**0**	0.0646	0.0938	0.2379	1.0031
Med	1.72E-03	0	**0**	0.0880	0.2373	0.1724	3.9173
Time	3.7118	3.0492	2.8285	0.3632	0.5784	1.5625	**0.2827**
f_3 Ave	2.70E-04	2.01E-04	**0**	7.8103	22.489	555.90	411.12
SD	1.48E-03	8.54E-04	**0**	9.7981	14.114	250.69	171.09
Med	7.83E-04	3.58E-04	**0**	4.9511	19.099	545.68	406.06
Time	3.1909	3.1479	2.8685	0.3628	0.6102	1.6234	**0.2861**
f_4 Ave	0	0	**0**	1.1526	2.6088	1.5937	15.232
SD	0	0	**0**	0.6140	0.8389	1.2135	2.8518
Med	0	0	**0**	1.0469	2.4961	1.7188	14.820
Time	3.0484	2.9475	2.8848	0.3685	0.6060	1.6525	**0.3000**
f_5 Ave	3.1111	2.9961	**2.6767**	25.074	148.08	369.75	83.19
SD	0.3422	0.5206	**0.2974**	28.443	137.19	342.88	155.51
Med	3.5845	3.1526	**2.8178**	14.932	96.094	305.55	125.23
Time	2.9038	2.8007	2.8109	0.3389	0.5702	1.5375	**0.2671**
f_6 Ave	0.5738	0.3360	**0.2241**	2.6993	8.4966	6.9842	435.33
SD	0.1550	0.1667	**0.1586**	7.0104	6.1409	7.0104	247.11
Med	0.6125	0.4265	**0.3021**	1.5588	7.6725	4.6712	440.29
Time	2.9606	2.8254	2.8836	0.3503	0.5763	1.5683	**0.2760**
f_7 Ave	3.61E-04	1.80E-04	**9.14e-5**	0.0060	0.0155	0.0472	3.22E-03
SD	2.70E-04	1.51E-04	**1.21e-4**	0.0044	0.0075	0.0436	2.22E-03
Med	3.98E-04	3.02E-04	**4.22e-5**	0.0057	0.0140	0.0348	2.83E-03
Time	0.9189	0.90379	0.74669	**0.1914**	0.2772	1.0258	0.2033

Simulations for Multimodal Benchmark Functions

A total of seven multimodal benchmark functions are introduced to verify the effectiveness of the proposed algorithm, HBBA. The simulation results are shown in Table 8, and the convergence curves of best solutions for functions f_8 to f_{14} are given in Figure 7.

Figure 5. Convergence curves of best solutions for unimodal function ($f_1 - f_4$) (Red-HBBA, Black-BBA, Blue-BPSO, Green-GA)

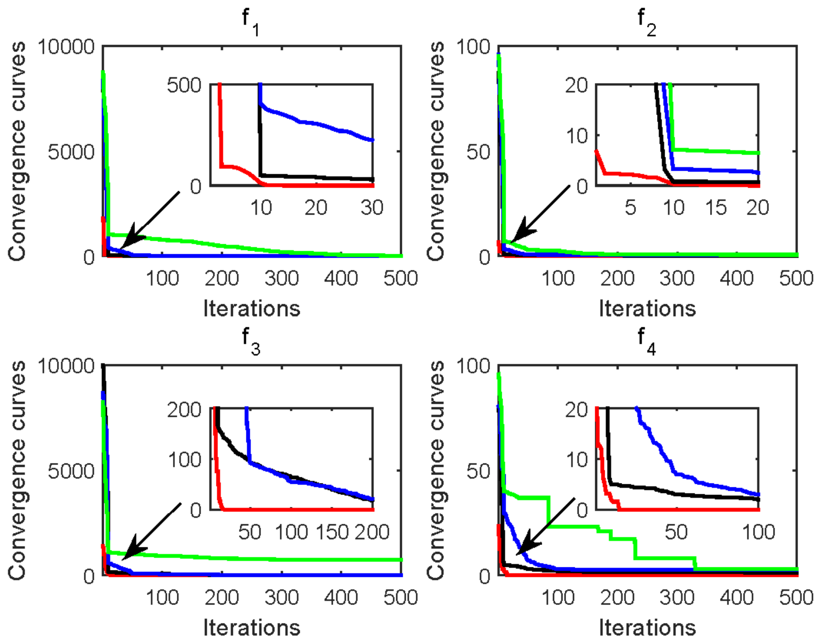
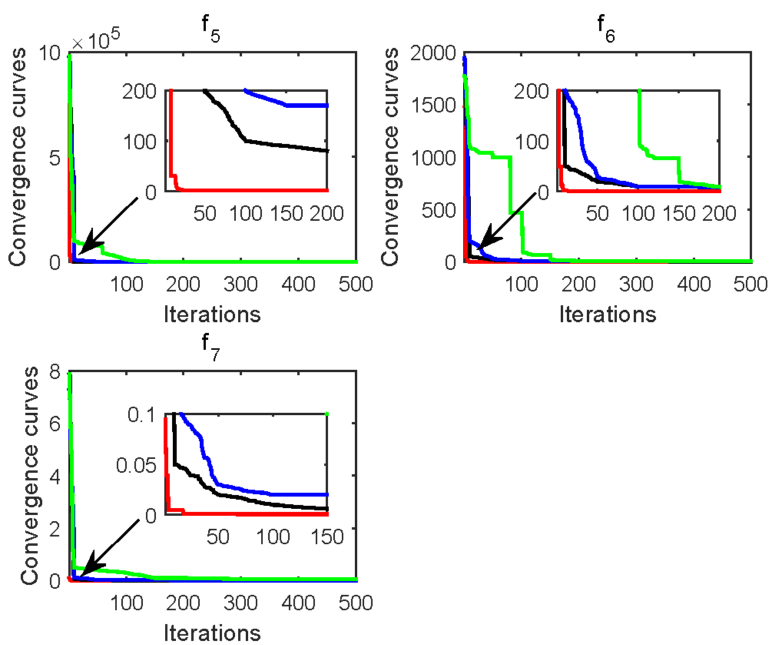


Figure 6. Convergence curves of best solutions for unimodal function ($f_5 - f_7$) (Red-HBBA, Black-BBA, Blue-BPSO, Green-GA)



The comparison results are given in Table 8. It can be seen that HBBA achieves all the best solutions in average, standard deviation, and median for the seven multimodal benchmark functions. For functions f_9, f_{11}, and f_{14}, the real global optimal solutions (i.e., 0, 0, -1) are all obtained by HBBA. For function f_{10}, the global optimal solution generated by HBBA is about 8.882e-16 in all 30 independent runs, which is very close to the real global optimal solution 0. For BBA, BPSO, GA and binABC, the obtained optimal solutions are obviously bigger than those by HBBA for functions f_9, f_{10}, f_{11} and f_{14}. For function f_8, the medians of optimal solutions achieved by the HBBA and other four comparison algorithms are -1903.24, -994.8, -992.4, -918.6 and -1547.78, respectively, and the real minimum value of f_8 is -2094.9145. For function f_{12}, HBBA ranks in the top two for median value in the four algorithms and obtains best average values. As for function f_{13}, the average and median values obtained by HBBA are very close to the corresponding ones by BBA. More importantly, the real optimal solution of f_{13} (i.e., -1) is hit by 15 counts in total 30 independent runs by HBBA. Therefore, HBBA gets much better solutions compared to BBA, BPSO, GA and binABC for multimodal benchmark functions. In addition, it can be seen that the best average and median values are obtained by HBBA compared to BBA,HBBA_1 and HBBA_2 except f_{13}. This is mainly because the update in unknown spaces for each dimensional component of velocity vector separately is realized by the modified random black hole model, and the chaotic map effectively avoids the premature convergence problem.

The convergence curves of best solutions obtained by HBBA are shown in Figure 7. Among them, the numbers of the convergence steps of functions f_9, f_{10}, f_{11}, and f_{14} are 38, 15, 21, and 17, respectively, while the corresponding numbers of convergence steps obtained by BBA are about 375, 395, 430, and 400, respectively (Mirjalili et al., 2014). For functions f_8, f_{12}, and f_{13}, the final optimal solutions obtained by HBBA are obviously superior to other solutions. Especially, for functions f_8, f_9, f_{11}, f_{13}, and f_{14}, the real global optimal solutions are found by HBBA, which are 0, 0, 0, -1, and -1, respectively.

Therefore, whether on the optimal solutions or on the convergence processes, HBBA achieves much better performance compared to the other three algorithms for the above multimodal benchmark functions.

Simulations for Composition Benchmark Functions

Several composition benchmark functions (Mirjalili et al., 2014) are introduced to verify the performance of HBBA in this subsection. This type of benchmark function is composed of different compositions including sphere function, Griewank's function, Rastrigin's function, Weierstrass's function, and Ackley's function. The different compositions increase the difficulty for algorithms to find the optimal solutions. To effectively deal with this problem, the proposed time-varying V-shaped transfer function is applied to this type of function to help find more accurate solutions.

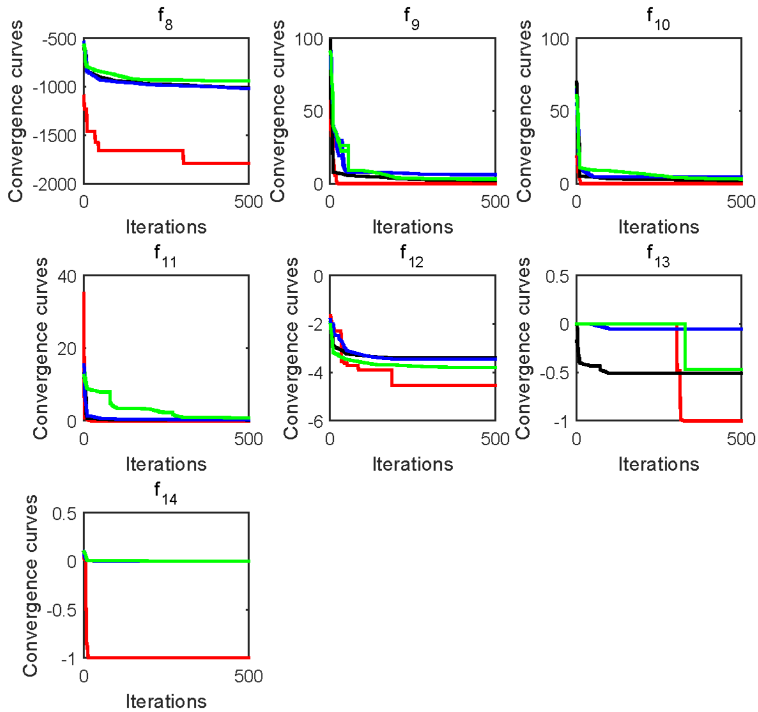
The simulation results are given in Table 9. It can be seen that HBBA achieves much better solutions for functions f_{15}, f_{16} and f_{19}. The medians for f_{15}, f_{16}, and f_{19} are about 6.0678, 36.5638, and 24.4591, respectively, while the corresponding medians obtained by BBA are about 78.7049, 154.589, and 163.111, which are nearly 13, 4.2, and 6.7 times of the solutions obtained by HBBA, respectively. For function f_{17}, although BBA gets the smallest average value, the difference between HBBA and BBA is only about 1.3440 (150.985-149.641). More important, the median obtained by HBBA (i.e., 142.896) is smaller than the corresponding value (i.e., 152.153) obtained by BBA. Therefore, for function f_{17}, the quality of solutions in total 30 independent runs obtained by HBBA

Table 8. Simulation results for multimodal benchmark functions in 30 independent runs

Function	HBBA_1	HBBA_2	HBBA	BBA	BPSO	GA	binABC
f_8 Ave	-1474.86	-1843.17	**-1959.67**	-985.3	-988.4	-929.3	-1565.13
SD	153.92	114.79	88.046	27.579	**14.219**	27.952	84.107
Med	-1428.75	-1835.25	**-1903.24**	-994.8	-992.4	-918.6	-1547.78
Time	3.8306	3.0546	2.9468	**0.3655**	0.6061	1.6854	0.4204
f_9 Ave	0.0367	6.46E-07	**0**	1.5850	4.9777	2.1896	18.145
SD	0.1836	3.54E-06	**0**	1.3353	2.5979	1.8330	3.5326
Med	0.0258	5.28E-07	**0**	1.2682	5.2827	1.9902	18.473
Time	3.0147	2.9169	2.9226	**0.3508**	0.5974	1.5962	0.3627
f_{10} Ave	8.882e-16	8.882e-16	**8.882e-16**	1.1560	2.7256	1.3999	10.598
SD	0	0	**0**	0.7179	0.1302	1.3381	1.2345
Med	8.882e-16	8.882e-16	**8.882e-16**	0.9589	0.3862	2.3168	10.920
Time	3.2138	2.9286	2.8846	0.3669	0.6282	1.6952	**0.2762**
f_{11} Ave	8.39E-06	0	**0**	0.2463	0.3873	0.7067	4.8262
SD	4.59E-05	0	**0**	0.0839	0.1302	0.3223	1.7503
Med	6.35E-06	0	**0**	0.2261	0.3862	0.7336	4.8367
Time	3.2932	2.8535	2.8563	0.3546	0.5968	1.5923	**0.2670**
f_{12} Ave	-3.2917	-4.0031	**-4.4813**	-3.6425	-3.6416	-3.8849	-2.5761
SD	0.3407	0.3283	**0.2031**	0.3506	0.3245	0.7177	0.2842
Med	-3.5228	-3.9585	-4.03125	-3.6080	-3.5816	**-4.0761**	-2.5330
Time	3.0262	2.8708	2.8841	0.3629	0.6228	1.6563	**0.2943**
f_{13} Ave	3.06E-25	-0.40388	-0.502	**-0.5173**	-0.0555	-0.4746	3.71E-21
SD	1.63E-24	0.42585	0.5066	0.3841	**0.1351**	0.4856	7.57E-21
Med	2.57E-25	-0.4357	-0.5300	**-0.5908**	-4e-109	-0.4141	2.49E-22
Time	2.9542	3.0037	3.0434	0.3744	0.6747	1.7358	**0.3612**
f_{14} Ave	-0.3228	-0.9671	**-1**	3.198e-4	2.95e-4	0.0016	0.0053
SD	0.4564	0.0901	**0**	2.334e-4	0.00021	0.0008	0.0025
Med	-0.6854	-0.9235	**-1**	2.325e-4	0.00027	0.0013	0.0052
Time	2.8786	3.0664	**3.3025**	0.3683	0.6298	1.6352	0.2825

is superior to the solutions obtained by BBA. For functions f_{18} and f_{20}, HBBA gets second in the four comparison algorithms. In addition, HBBA_2 achieves much better solutions than that of HBBA_1. This is mainly because of the fact that time-varying V-shaped transfer function provides

Figure 7. Convergence curves of best solutions for multimodal function ($f_8 - f_{14}$) (Red-HBBA, Black-BBA, Blue-BPSO, Green-GA)



a faster switching speed for position in binary space. Nevertheless, the solution obtained by HBBA incorporating chaotic map is better than that of HBBA_2. Totally, in the six composition benchmark functions, HBBA gets best solutions in four functions, and the effectiveness of HBBA in dealing with composition benchmark functions is verified.

The convergence curves of best solutions in 30 independent runs for f_{15} - f_{20} are shown in Figure 8. For functions f_{15}, f_{16}, f_{17}, and f_{19}, the convergence curves generated by HBBA are significantly better than the curves generated by other three algorithms. This also demonstrates the superiority of the proposed algorithm.

In summation, in a total of 20 benchmark functions, including unimodal, multimodal, and composition functions, HBBA obtains better solutions in 16 functions in terms of average value, standard deviation, and median value. For the rest four functions, HBBA ranks in the top two among the four algorithms for all the four functions. This achievement announces the superiority of HBBA.

Unit Commitment Problem

In modern power systems, generation sources involve thermal, wind, photovoltaic, and hydroelectric power. To ensure the safety and economy of power generation in power systems, unit commitment plays an important role. The unit commitment problem (UCP) is a constrained combinatorial optimization problem that minimizes power generation cost by determining on/off states of units. Under different system and operation constraints, the minimum generator cost is to minimize the start-up cost, shutdown cost, and operation cost (Pan et al., 2021). The mathematical model of UCP is achieved as follows:

Table 9. Simulation results for composition benchmark functions in 30 independent runs

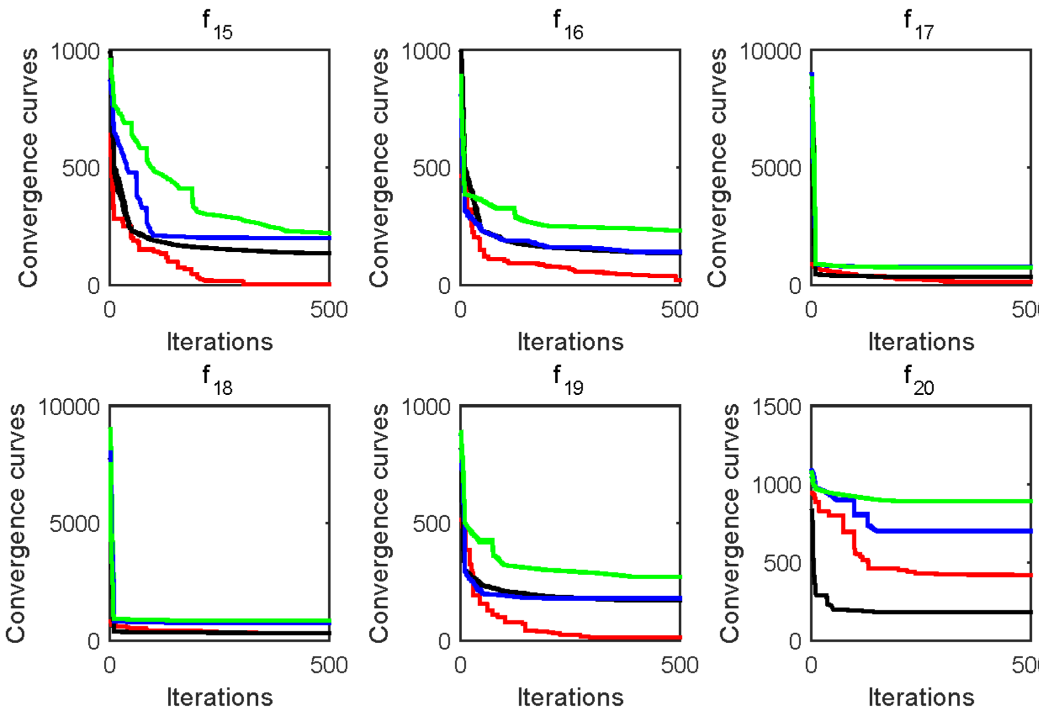
Function	HBBA_1	HBBA_2	HBBA	BBA	BPSO	GA	binABC
f_{15} Ave	148.080	17.4913	**14.4485**	93.2475	194.852	193.668	369.94
SD	27.3737	16.8622	**13.8159**	64.2902	60.0340	121.913	57.39
Med	148.639	9.5673	**6.0678**	78.7049	176.038	170.521	321.75
Time	29.282	25.9335	29.6994	1.0244	1.0478	3.25874	**0.7059**
f_{16} Ave	118.823	49.8088	**36.8855**	156.632	146.761	205.679	696.78
SD	8.6719	9.3764	**7.4168**	31.8847	29.0801	160.985	63.13
Med	118.428	46.6513	**36.5638**	154.589	140.642	154.868	638.45
Time	27.1188	26.4750	26.3512	1.0095	1.0626	3.29875	**0.7273**
f_{17} Ave	538.912	180.266	150.985	**149.641**	445.776	384.776	726.58
SD	59.025	26.1351	**24.9431**	38.7091	49.3449	118.031	60.58
Med	516.283	184.015	**142.896**	152.153	443.398	448.191	657.28
Time	26.2933	27.0835	26.1341	1.0606	1.1056	3.35786	**0.9539**
f_{18} Ave	614.693	354.936	345.005	**146.948**	479.987	588.126	722.61
SD	65.5889	22.4622	**11.6124**	22.9687	30.194	102.337	64.67
Med	633.192	347.136	344.472	**147.049**	477.019	639.901	705.65
Time	29.95	33.8932	29.7607	**1.0393**	1.0724	3.15857	1.1433
f_{19} Ave	158.089	24.5026	**24.1934**	166.121	172.082	246.302	604.26
SD	13.408	7.5069	**5.9879**	49.8056	64.2674	183.534	52.72
Med	156.185	24.6721	**24.4591**	163.111	140.993	218.013	589.78
Time	29.9704	29.7138	34.1758	1.0120	1.0594	3.04582	**0.8889**
f_{20} Ave	589.908	496.565	485.319	**152.812**	691.650	914.537	610.83
SD	21.2738	38.9557	37.8245	**33.6342**	149.626	12.3219	57.06
Med	588.957	513.176	510.032	**145.539**	607.977	908.362	582.39
Time	29.8742	30.5417	30.7456	1.0124	1.0649	3.08324	**0.6916**

Objective Function

The objective of UC aims to minimize the fuel cost and start-up cost when meeting system and operation constraints. The fuel cost and start-up cost are provided as (17) and (18):

$$Fitness = \min \sum_{t=1}^{T} \sum_{i=1}^{N} C_f \left(P_i^t \right) S_i^t + C_{it}^t S_i^t \tag{16}$$

Figure 8. Convergence curves of best solutions for composite function ($f_{15} - f_{20}$) (Red-HBBA, Black-BBA, Blue-BPSO, Green-GA)



$$C_f\left(P_i^t\right) = a_i + b_i\left(P_i^t\right) + c_i\left(P_i^t\right)^2 \tag{17}$$

$$C_{\mathrm{it}}^t = \begin{cases} C_{it}^{\mathrm{hot}}, & T_{\mathrm{down},i}^{\min} \le T_{\mathrm{off},i} \le \left\{\mathrm{Cold}_i + T_{\mathrm{down},i}^{\min}\right\} \\ C_{it}^{\mathrm{cold}}, & \left\{\mathrm{Cold}_i + T_{\mathrm{down},i}^{\min}\right\} \le T_{\mathrm{off}\,,i} \end{cases} \tag{18}$$

where S_i^t is the on/off status of unit i at time t; $C_f\left(P_i^t\right)$ achieves total fuel cost; P_i^t is the active power of unit i at time t; C_{it}^t is the start-up cost at time t; T and i are the number of scheduling time and generation unit; and a_i, b_i, c_i are respectively the cost coefficients of units; C_{it}^{hot} and C_{it}^{cold} imply hot start-up cost and cold start-up cost, respectively; $T_{\mathrm{off},i}$ represents de-committed time of unit i; Cold_i is the cold start hours of unit i and $T_{\mathrm{down},i}^{\min}$ is the minimum time between two consecutive commitment generation units.

Constraints

1. *Load constraints:* The total generator power from all committed units must meet the hourly load demand, as following:

$$\sum_{i=1}^{N} P_i^t S_i^t = PD_{\mathrm{sys}}^t \qquad \forall t \in T \tag{19}$$

where PD_{sys}^t represents the total generation load demand at hour t.

2. *Spinning reserve constraint:* It is necessary to immediately offer redundant power generation, known as spinning reserve, which is triggered by the failure of the working unit or an unexpected surge in load demand. The spinning reserve is expressed as:

$$\sum_{i=1}^{N} S_i^t P_{i(\max)} \geq PD_{sys}^t + P_R^t \tag{20}$$

where P_R^t implies the spinning reserve at hour t.

3. *Generation power limits:* Each generation unit in committed has a output limit, which is given by,

$$S_i^t P_{i(\min)} \leq P_i^t \leq S_i^t P_{i(\max)} \tag{21}$$

where $P_{i(\min)}$ and $P_{i(\max)}$ are minimum and maximum generation limit of unit i, respectively.

4. *Minimum up/down time constraints:* Because of the minimum up/down time constraint, once each unit is online/offline, it cannot be immediately shut down/started. The minimum up/down time constraints are expressed as:

$$S_i^t = \begin{cases} 1 & \text{if } S_i^{t+1} \quad T_{\mathrm{down},i}^{\min} \leq T_{\mathrm{off},i} \\ 0 & \text{if } (1 - S_i^{t+1}) \; T_{\mathrm{up},i}^{\min} \leq T_{\mathrm{off},i} \end{cases} \tag{22}$$

where S_i^{t+1} is the status of unit i at hour $t+1$.

In practice, UCP is a combinatorial optimization problem including discrete variables and continuous variables, namely, 'determining the generator state' and 'economic dispatch'. The proposed HBBA is employed to determine the schedule state of generation in binary space, and the Lambda-iteration method (Singhal et al., 2014) is adopted to solve economic dispatch in continuous space. These test systems achieve 10-unit, 100-unit. The load demand and characteristics of 10-unit are taken from Kamboj et al. (2016). The algorithm parameters are the same as the above in Section 4.

To verify the effectiveness of the proposed HBBA, the comparison of both HBBA and other well-known methods for 10 units UCP is provided as Table 10. The comparison methods have EP (Juste et al., 1999), GA (Kazarlis et al., 1996), SA (Simopoulos et al., 2006), BPSO (Z. Yang et al., 2017), BLPSO (Z. Yang et al., 2017), Enh-hGADE (Trivedi et al., 2015), iDA-PSO (Khunkitti et al., 2019), BFMO (Pan et al., 2021), BCSO (Y. Wang et al., 2019) and BBA (Reddy et al., 2017). Form Table 10, HBBA and BBA have the lowest cost and the same value as all of the compared algorithms (563937.307 \$). The HBBA solution outperforms the BBA solution in terms of both average and maximum cost. Furthermore, the average execution time of HBBA for 30 independent runs is 19.068 s, which is 24.1% faster than that of BBA. When compared to other algorithms, HBBA performs well

Table 10. Comparison cost results of HBBA for 10-Unit test problem

Algorithm	Minimum ($)	Average ($)	Maximum ($)	Time (s)
EP	564551	565352	566231	100
GA	565825	-	570032	221
SA	565828	565988	566260	3.35
BPSO	563955.99	564000.40	564053.73	25.45
BLPSO	563977.01	563982.09	563987.16	22.09
iDA-PSO	565807.309	565827.014	565891.759	231.31
Enh-hGADE	563938	563997	564261	26
BFMO	-	564864	-	-
BBA	563937.687	564568.853	565205.721	25.130
HBBA	563937.307	563976.735	564036.467	19.068

in terms of minimum, average, maximum, and average running time. For example, HBBA performs best in terms of minimum and average values (563937.307 $, 563976.735 $). HBBA ranks second among all pairs of algorithms in terms of maximum value and average running time. Based on the above analysis, the superiority of HBBA in the application of the power system unit commitment problem is verified.

To further illustrate the effectiveness of HBBA on large-scale UCP problems, 100-unit is examined, as shown in Table 11. From Table 11, it can be clearly seen that the best solutions in terms of minimum, average, and maximum are obtained by HBBA. Specifically, the minimum obtained by HBBA is able to save 8140.609$, 17701.609$ and 14149.609$ per day, compared to the minimum obtained by EP, GA, and SA. In addition, the average execution time of EP, GA, and SA are nearly 20, 51, and 2.2 that of HBBA, respectively. For BCSO, BPSO, BLPSO, BGWO and BFMO, the solutions obtained by HBBA are best. As for BBA, the average cost of HBBA is 96.8% of that of BBA, despite the fact that the average execution time of HBBA is longer. In conclusion, the effectiveness of HBBA is validated for large-scale UCP problems.

Table 11. Comparison cost results of HBBA for 100-unit test problem

Algorithm	Minimum ($)	Average ($)	Maximum ($)	Time (s)
EP	5623885	5633800	5639148	6120
GA	5627437	-	5637914	15733
SA	5617876	5624301	5628506	696
BCSO	5610281.71	5610624.74	5610986.92	85.01
BPSO	566545.61	5482671.50	5692414.71	112.45
BLPSO	5655610.14	5655610.14	5655610.14	113.54
BGWO	5628975	5637659	5643899	836.54
BFMO	-	564864	-	-
BBA	5787294.136	5796318.181	5803759.278	179.162
HBBA	5609735.391	5614292.836	5618754.891	305.208

CONCLUSION

This paper gives a new hybrid binary bat algorithm (HBBA) by integrating a modified random black hole model, and a time-varying V-shaped transfer function is also proposed to further improve the performance of HBBA. The superiority of the proposed HBBA is verified by three types of benchmark functions and an engineering optimization problem. For further studies, different applications by HBBA can be investigated such as feature selection and mixed-integer linear programming. The modified random black hole model and the proposed time-varying V-shaped transfer function can also be applied to other heuristic algorithms, and this is left as an interesting future work.

CONFLICTS OF INTEREST

The authors of this publication declare that there are no competing interests.

FUNDING STATEMENT

This research was supported by the National Natural Science Foundation of China [grant number 62163013].

PROCESS DATES

Received: August 10, 2023, Revision: January 8, 2024, Accepted: January 8, 2024.

CORRESPONDING AUTHOR

Correspondence should be addressed to: Huijun Liang, lhj@hbmzu.edu.cn

REFERENCES

Abarajithan, S., & Vijayarani, S. (2022). A hybrid harmony search: Antlion optimization algorithm for multi-level high utility association rule mining. [IJSIR]. *International Journal of Swarm Intelligence Research*, *13*(1), 1–20. doi:10.4018/IJSIR.308293

Abdel-Basset, M., Mohamed, R., & Mirjalili, S. (2021). A binary equilibrium optimization algorithm for 0–1 knapsack problems. *Computers & Industrial Engineering*, *151*, 106946.. doi:10.1016/j.cie.2020.106946

Abualigah, L., Yousri, D., Abd Elaziz, M., Ewees, A. A., Al-Qaness, M. A., & Gandomi, A. H. (2021). Aquila optimizer: A novel meta-heuristic optimization algorithm. *Computers & Industrial Engineering*, *157*, 107250. doi:10.1016/j.cie.2021.107250

Akila, S., & Christe, S. A. (2022). A wrapper based binary bat algorithm with greedy crossover for attribute selection. *Expert Systems with Applications*, *187*, 115828. doi:10.1016/j.eswa.2021.115828

Al-Tashi, Q., Abdulkadir, S. J., Rais, H. M., Mirjalili, S., Alhussian, H., Ragab, M. G., & Alqushaibi, A. (2020). Binary multi-objective grey wolf optimizer for feature selection in classification. *IEEE Access : Practical Innovations, Open Solutions*, *8*, 106247–106263. doi:10.1109/ACCESS.2020.3000040

Aslan, M., Gunduz, M., & Kiran, M. S. (2019). JayaX: Jaya algorithm with xor operator for binary optimization. *Applied Soft Computing*, *82*, 105576. doi:10.1016/j.asoc.2019.105576

Bambi, C., Freese, K., Vagnozzi, S., & Visinelli, L. (2019). Testing the rotational nature of the supermassive object M87* from the circularity and size of its first image. *Physical Review. D*, *100*(4), 044057. doi:10.1103/PhysRevD.100.044057

Beheshti, Z. (2020). A time-varying mirrored S-shaped transfer function for binary particle swarm optimization. *Information Sciences*, *512*, 1503–1542. doi:10.1016/j.ins.2019.10.029

Dorigo, M., & Gambardella, L. M. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, *1*(1), 53–66. doi:10.1109/4235.585892

Du, X., Zhou, J., Ni, Y., Liu, W., Xiao, R., & Wu, X. (2023). A novel binary multi-swarms fruit fly optimisation algorithm for the 0-1 multidimensional knapsack problem. *International Journal of Bio-inspired Computation*, *21*(1), 1–10. doi:10.1504/IJBIC.2023.129982

El-Kenawy, E. S. M., Mirjalili, S., Alassery, F., Zhang, Y. D., Eid, M. M., El-Mashad, S. Y., Aloyaydi, B. A., Ibrahim, A., & Abdelhamid, A. A. (2022). Novel meta-heuristic algorithm for feature selection, unconstrained functions and engineering problems. *IEEE Access : Practical Innovations, Open Solutions*, *10*, 40536–40555. doi:10.1109/ACCESS.2022.3166901

Gupta, D., Arora, J., Agrawal, U., Khanna, A., & de Albuquerque, V. H. C. (2019). Optimized binary bat algorithm for classification of white blood cells. *Measurement*, *143*, 180–190. doi:10.1016/j.measurement.2019.01.002

Hashim, F. A., & Hussien, A. G. (2022). Snake optimizer: A novel meta-heuristic optimization algorithm. *Knowledge-Based Systems*, *242*, 108320. doi:10.1016/j.knosys.2022.108320

He, Y., Zhang, F., Mirjalili, S., & Zhang, T. (2022). Novel binary differential evolution algorithm based on Taper-shaped transfer functions for binary optimization problems. *Swarm and Evolutionary Computation*, *69*, 101022. doi:10.1016/j.swevo.2021.101022

Hematpour, N., & Ahadpour, S. (2021). Execution examination of chaotic S-box dependent on improved PSO algorithm. *Neural Computing & Applications*, *33*(10), 5111–5133. doi:10.1007/s00521-020-05304-9

Hichem, H., Elkamel, M., Rafik, M., Mesaaoud, M. T., & Ouahiba, C. (2022). A new binary grasshopper optimization algorithm for feature selection problem. *Journal of King Saud University. Computer and Information Sciences*, *34*(2), 316–328. doi:10.1016/j.jksuci.2019.11.007

Holland, J. H. (1992). Genetic algorithms. *Scientific American*, *267*(1), 66–73. doi:10.1038/scientificamerican0792-66

Juste, K. A., Kita, H., Tanaka, E., & Hasegawa, J. (1999). An evolutionary programming solution to the unit commitment problem. *IEEE Transactions on Power Systems*, *14*(4), 1452–1459. doi:10.1109/59.801925

Kamboj, V. K., Bath, S. K., & Dhillon, J. S. (2016). Implementation of hybrid harmony search/random search algorithm for single area unit commitment problem. *International Journal of Electrical Power & Energy Systems*, *77*, 228–249. doi:10.1016/j.ijepes.2015.11.045

Kazarlis, S. A., Bakirtzis, A. G., & Petridis, V. (1996). A genetic algorithm solution to the unit commitment problem. *IEEE Transactions on Power Systems*, *11*(1), 83–92. doi:10.1109/59.485989

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proc. of ICNN'95-international conference on neural networks,* (vol. 4, pp. 1942-1948). IEEE. doi:10.1109/ICNN.1995.488968

Khunkitti, S. R., Watson, N., Chatthaworn, R., Premrudeepreechacharn, S., & Siritaratiwat, A. (2019). An improved DA-PSO optimization approach for unit commitment problem. *Energies*, *12*(12), 2335. doi:10.3390/en12122335

Kiran, M. S. (2015). The continuous artificial bee colony algorithm for binary optimization. *Applied Soft Computing*, *33*, 15–23. doi:10.1016/j.asoc.2015.04.007

Liang, H., Liu, Y., Shen, Y., Li, F., & Man, Y. (2018). A hybrid bat algorithm for economic dispatch with random wind power. *IEEE Transactions on Power Systems*, *33*(5), 5052–5061. doi:10.1109/TPWRS.2018.2812711

Liang, J. J., Suganthan, P. N., & Deb, K. (2005, June). Novel composition test functions for numerical global optimization. In *Proceedings 2005 IEEE swarm intelligence symposium*, (pp. 68-75). IEEE. doi:10.1109/SIS.2005.1501604

Mirjalili, S., & Lewis, A. (2013). S-shaped versus V-shaped transfer functions for binary particle swarm optimization. *Swarm and Evolutionary Computation*, *9*, 1–14. doi:10.1016/j.swevo.2012.09.002

Mirjalili, S., Mirjalili, S. M., & Yang, X. S. (2014). Binary bat algorithm. *Neural Computing & Applications*, *25*(3-4), 663–681. doi:10.1007/s00521-013-1525-5

Nadimi-Shahraki, M. H., Taghian, S., Mirjalili, S., Zamani, H., & Bahreininejad, A. (2022). GGWO: Gaze cues learning-based grey wolf optimizer and its applications for solving engineering problems. *Journal of Computational Science*, *61*, 101636. doi:10.1016/j.jocs.2022.101636

Pan, J. S., Hu, P., & Chu, S. C. (2021). Binary fish migration optimization for solving unit commitment. *Energy*, *226*, 120329. doi:10.1016/j.energy.2021.120329

Pashaei, E., & Pashaei, E. (2022). An efficient binary chimp optimization algorithm for feature selection in biomedical data classification. *Neural Computing & Applications*, *34*(8), 6427–6451. doi:10.1007/s00521-021-06775-0

Qin, Q., Cheng, S., Chu, X., Lei, X., & Shi, Y. (2017). Solving non-convex/non-smooth economic load dispatch problems via an enhanced particle swarm optimization. *Applied Soft Computing*, *59*, 229–242. doi:10.1016/j.asoc.2017.05.034

Reddy, S., Kumar, R., & Panigrahi, B. K. (2017, December). Binary bat search algorithm for unit commitment problem in power system. In 2017 IEEE international WIE conference on electrical and computer engineering (WIECON-ECE) (pp. 121-124). IEEE.

Reddy, S., Panwar, L. K., Panigrahi, B. K., & Kumar, R. (2018). Solution to unit commitment in power system operation planning using binary coded modified moth flame optimization algorithm (BMMFOA): A flame selection based computational technique. *Journal of Computational Science*, *25*, 298–317. doi:10.1016/j.jocs.2017.04.011

Simopoulos, D. N., Kavatza, S. D., & Vournas, C. D. (2006). Unit commitment by an enhanced simulated annealing algorithm. *IEEE Transactions on Power Systems*, *21*(1), 68–76. doi:10.1109/TPWRS.2005.860922

Singhal, P. K., Naresh, R., Sharma, V., & Kumar, G. (2014, May). Enhanced lambda iteration algorithm for the solution of large scale economic dispatch problem. In *International conference on recent advances and innovations in engineering (ICRAIE-2014)* (pp. 1-6). IEEE. doi:10.1109/ICRAIE.2014.6909294

Suresh, S., & Lal, S. (2017). Multilevel thresholding based on chaotic Darwinian particle swarm optimization for segmentation of satellite images. *Applied Soft Computing*, *55*, 503–522. doi:10.1016/j.asoc.2017.02.005

Trivedi, A., Srinivasan, D., Biswas, S., & Reindl, T. (2015). Hybridizing genetic algorithm with differential evolution for solving the unit commitment scheduling problem. *Swarm and Evolutionary Computation*, *23*, 50–64. doi:10.1016/j.swevo.2015.04.001

Turkoglu, B., Uymaz, S. A., & Kaya, E. (2022). Binary artificial algae algorithm for feature selection. *Applied Soft Computing*, *120*, 108630. doi:10.1016/j.asoc.2022.108630

Usman, A. M., Yusof, U. K., & Naim, S. (2022). Filter-based feature selection: A comparison among binary and continuous Cuckoo optimisation algorithms along with multi-objective optimisation algorithms using gain ratio-based entropy. *International Journal of Bio-inspired Computation*, *20*(3), 183–192. doi:10.1504/IJBIC.2022.127511

Wang, G. G., Gao, D., & Pedrycz, W. (2022). Solving multiobjective fuzzy job-shop scheduling problem by a hybrid adaptive differential evolution algorithm. *IEEE Transactions on Industrial Informatics*, *18*(12), 8519–8528. doi:10.1109/TII.2022.3165636

Wang, Y., Yang, Z., Guo, Y., Zhou, B., & Zhu, X. (2019). A novel binary competitive swarm optimizer for power system unit commitment. *Applied Sciences (Basel, Switzerland)*, *9*(9), 1776. doi:10.3390/app9091776 PMID:32754346

Yang, X. S. (2010a). A new metaheuristic bat-inspired algorithm. In *Nature inspired cooperative strategies for optimization (NICSO 2010)* (pp. 65–74). Springer Berlin Heidelberg. doi:10.1007/978-3-642-12538-6_6

Yang, Z., Li, K., Niu, Q., & Xue, Y. (2017). A novel parallel-series hybrid meta-heuristic method for solving a hybrid unit commitment problem. *Knowledge-Based Systems*, *134*, 13–30. doi:10.1016/j.knosys.2017.07.013

Zhang, H. (2020). A binary cooperative bat algorithm based optimal topology design of leader–follower consensus. *ISA Transactions*, *96*, 51–59. doi:10.1016/j.isatra.2019.06.010 PMID:31221466

Zhang, J., Liu, K., Tan, Y., & He, X. (2008, June). *Random black hole particle swarm optimization and its application. In 2008 international conference on neural networks and signal processing*. IEEE.

Huijun Liang received the Ph.D. degree in control theory and control engineering from Shandong University, Jinan, China, in 2020. He is currently a lecture with the College of Intelligent Systems Science and Engineering, Hubei Minzu University, Enshi China. His current research interests include computational intelligence and optimization in power systems.

Aokang Pang received the B.Eng. from Xi'an Kedagaoxin University, Xi'an, China, in 2021. Currently, he is a M.Eng. degree candidate student in the College of Intelligent Systems Science and Engineering, Hubei Minzu University, Enshi, China. His current research interests include application of computational intelligence and optimization in power systems.

Chenhao Lin received the B.E. degree in electrical engineering and its automation from North China Electric Power University, Baoding, China, in 2018. He is currently pursuing the M.E. degree in electrical engineering in Hubei Minzu University, Enshi, China. His current research interests include computational intelligence and optimization in power systems.

Jianwei Zhong received the M.E. degree in material processing engineering from Huazhong University of Science and Technology, Wuhan, China, in 2004. He is currently working as a Professor in Hubei Minzu University, Enshi, China. His current research interests include power system operation and new energy generation.